Applied Mathematical Sciences, Vol. 7, 2013, no. 24, 1199 - 1203
HIKARI Ltd, www.m-hikari.com

The Identity Checking Problem for Semigroups

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
gorbenko.ann@gmail.com

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

In this paper, we consider the identity checking problem for semigroups. We propose a genetic algorithm to solve the problem.

Keywords: identity checking problem, semigroup, genetic algorithm

There is a considerable interest in investigation of semigroup identities (see e.g. [1] – [4]). In particular, the identity checking problem for finite semigroups is extensively studied (see e.g. [5] and references in [5]). The identity checking problem in semigroup $\mathcal{A}$ is the following combinatorial decision problem.

CHECK-ID($\mathcal{A}$):

INSTANCE: *Words of variables u and v.*

QUESTION: *Whether or not the identity $u = v$ holds in $\mathcal{A}$?*

Usually, a semigroup for an instance of the identity checking problem for finite semigroups is given by a semigroup multiplication table. There is a finite semigroup $\mathcal{A}$ such that CHECK-ID($\mathcal{A}$) is **co-NP**-complete (see e.g. [5]).

Note that a large number of algorithmic problems of robotics received a lot of attention recently (see e.g. [6] – [15]). The representation of robotic systems plays an important role in solutions of robotic tasks (see e.g. [16]). There is a natural way to represent a robotic system by elements of some semigroup (see e.g. [17]). But in this case, we need to consider exponentially large or infinite semigroups. Therefore, for robotic systems, a representation by semigroup relations is preferred. It should be noted that in some cases a representation by relations and identities may be considerably shorter than any representation by relations. So, for robotic systems, it is intresting to consider the problem CHECK-ID($\mathcal{A}$) where $\mathcal{A}$ is a description of some robotic system. In this paper, we assume that a semigroup is given by a set of semigroup relations. In this case, also there is a finite semigroup $\mathcal{A}$ such that CHECK-ID($\mathcal{A}$) is **co-NP**-complete. There is a infinite semigroup $\mathcal{A}$ such that CHECK-ID($\mathcal{A}$) is undecidable (see e.g. [18]). In this paper, we consider a genetic algorithm to solve the problem.

Let Σ be a finite system of semigroup relations. Let

$$\mathcal{A} = \langle a[1], \ldots, a[n] \mid \Sigma \rangle$$

be a semigroup where $\{a[1], \ldots, a[n]\}$ is a set of generators of $\mathcal{A}$. Let

$$u(x[1], \ldots, x[m]) = v(x[1], \ldots, x[m])$$

be an identity.

To solve CHECK-ID($\mathcal{A}$), we can either derive the identity or to prove its falsity. Therefore, we use a parallel run of two sequences of genetic algorithms.

Assumption 0: $\mathcal{A} \not\models u(x[1], \ldots, x[m]) = v(x[1], \ldots, x[m])$**.**

- A genetic algorithm GA[0,1] for selection a set S of elements of $\mathcal{A}$.
- A genetic algorithm GA[0,2] for construction of homomorphism

$$H : \mathcal{A} \to \mathcal{S} = \langle S \mid \Sigma \rangle$$

  where $\mathcal{S}$ is the subsemigroup of $\mathcal{A}$ generated by S.
- A genetic algorithm GA[0,3] for construction of multiplication table of $\mathcal{S}$.
- A genetic algorithm GA[0,4] for selection a set F of elements of $\mathcal{S}$.

Consecutive run of genetic algorithms GA[0,1], GA[0,2], GA[0,3], GA[0,4] allows us to select values of $x[1], \ldots, x[m]$ that can potentially falsify the identity.

Assumption 1: $\mathcal{A} \models u(x[1], \ldots, x[m]) = v(x[1], \ldots, x[m])$**.**

- A genetic algorithm GA[1,1] for selection a set of templates

$$W = \{w \mid w \in (\{a[1], \ldots, a[n]\} \cup \{y[i] \mid i \in N\})^+\}.$$

- A genetic algorithm GA[1,2] for discovery a set of equalities

$$E = \{w[1] = w[2] \mid w[1], w[2] \in W, \mathcal{A} \models w[1] = w[2]\}$$

  where we consider

$$\{y[i] \mid i \in N\}$$

  as the set of variables.

- A genetic algorithm GA[1,3] for deduction a set of identities T.

- A genetic algorithm GA[1,4] for deduction

$$u(x[1], \ldots, x[m]) = v(x[1], \ldots, x[m]).$$

At first, we run GA[1,1] and create a set W. After this, we run GA[1,2]. GA[1,2] uses auxiliary genetic algorithm for initial prediction of elements of E. For initial value of E, we use a recursive parallel run of a genetic algorithm GA[1,4] for deduction of equalities and consecutive run of genetic algorithms GA[0,1], GA[0,2], GA[0,3], GA[0,4]. GA[1,4] allows us to prove some equalities from initial set E. Consecutive run of genetic algorithms GA[0,1], GA[0,2], GA[0,3], GA[0,4] we use to falsify some elements of E. GA[1,3] uses auxiliary genetic algorithm for initial prediction of elements of T. For initial value of T, we use a recursive run of GA[1,4] for deduction of identities. GA[1,4] uses four additional operators.

- Union of constants: if

$$\mathcal{A} \models w[1](x) = w[2](x),$$

  for any value of a constant x, then we can consider x as a variable.

- Separation of variables: if w is a some word, x is a variable, and $x \in w$, then we can replace x by any element of $\mathcal{A}$.

- Multiplication: for any $w[1]$ and $w[2]$, we can consider $w[1]w[2]$.

- Substitution: for any variable x and for any $w(x)$ and u, we can consider $w(u)$.

It is easy to see that we can use only GA[1,4] instead of usage of GA[1,1], GA[1,2], GA[1,3], and GA[1,4]. Selected experimental results are given in Table 1.

In Table 1, we assume equal allocation of computing resources for the consideration of assumptions 0 and 1. A[1] is an algorithm that uses only GA[1,4] after 10^5 generations. A[2] is an algorithm that uses GA[1,1], GA[1,2], GA[1,3],

	A[1]	A[2]	A[3]	A[4]
average time	94.17 h	72.89 h	61.75 h	42.3 h

Table 1: Experimental results for different genetic algorithms.

and GA[1,4] after 10^3 generations. A[3] is an algorithm that uses GA[1,1], GA[1,2], GA[1,3], and GA[1,4] after 10^4 generations. A[4] is an algorithm that uses GA[1,1], GA[1,2], GA[1,3], and GA[1,4] after 10^5 generations.

Since we use a parallel run of two sequences of genetic algorithms for the consideration of assumptions 0 and 1, we need some procedure to divide computing resources. In our computational experiments, we consider assumptions 0 and 1 on equal computing resources. Also, we consider a genetic algorithm for dynamic allocation of computing resources. Selected experimental results are given in Table 2.

	ER	DA(10^3)	DA(10^4)	DA(10^5)	DA(10^6)
average time	42.3 h	39.57 h	33.71 h	11.84 h	11.36 h

Table 2: Experimental results for resource allocation where ER denotes equal computing resources and DA(g) denotes dynamic allocation of computing resources with a genetic algorithm after g generations.

ACKNOWLEDGEMENTS. The work was partially supported by Analytical Departmental Program "Developing the scientific potential of high school" 8.1616.2011.

References

[1] V.Yu. Popov, On the finite base property for semigroup varieties, *Siberian Mathematical Journal*, 43 (2002), 910-919.

[2] V.Yu. Popov, The property of having independent basis in semigroup varieties, *Algebra and Logic*, 44 (2005), 46-54.

[3] V.Yu. Popov, Equational theories for classes of finite semigroups, *Algebra and Logic*, 40 (2001), 55-66.

[4] V.Yu. Popov, Decidability of equational theories of coverings of semigroup varieties, *Siberian Mathematical Journal*, 42 (2001), 1132-1141.

[5] J. Almeida, M.V. Volkov, and S.V. Goldberg, Complexity of the identity checking problem for finite semigroups, *Journal of Mathematical Sciences*, 158 (2009), 605-614.

[6] A. Gorbenko and V. Popov, On the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012) 1117-1120.

[7] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The problem of sensor placement for triangulation-based localisation, *International Journal of Automation and Control*, 5 (2011) 245-253.

[8] A. Gorbenko and V. Popov, Clustering Algorithm in Mobile Ad Hoc Networks, *Advanced Studies in Theoretical Physics*, 6 (2012) 1239-1242.

[9] A. Gorbenko and V. Popov, Multi-agent Path Planning, *Applied Mathematical Sciences*, 6 (2012) 6733-6737.

[10] A. Gorbenko and V. Popov, Programming for Modular Reconfigurable Robots, *Programming and Computer Software*, 38 (2012) 13-23.

[11] A. Gorbenko, M. Mornev, and V. Popov, Planning a Typical Working Day for Indoor Service Robots, *IAENG International Journal of Computer Science*, 38 (2011) 176-182.

[12] A. Gorbenko, V. Popov, and A. Sheka, Localization on Discrete Grid Graphs, *Lecture Notes in Electrical Engineering*, 107 (2012), 971-978.

[13] A. Gorbenko and V. Popov, The Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012) 4729-4732.

[14] A. Gorbenko and V. Popov, On the Problem of Placement of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012) 689-696.

[15] A. Gorbenko and V. Popov, Footstep Planning for Humanoid Robots, *Applied Mathematical Sciences*, 6 (2012) 6567-6571.

[16] A. Gorbenko, A. Lutov, M. Mornev, and V. Popov, Algebras of Stepping Motor Programs, *Applied Mathematical Sciences*, 5 (2011), 1679-1692.

[17] V. Popov, Independent Systems of Semigroup Relations and Descriptions of Robotic Systems, *Applied Mathematical Sciences*, 6 (2012) 6725-6727.

[18] V. Popov, The word problem for relatively free semigroups, *Semigroup Forum*, 72 (2006), 1-14.

Received: December 3, 2012